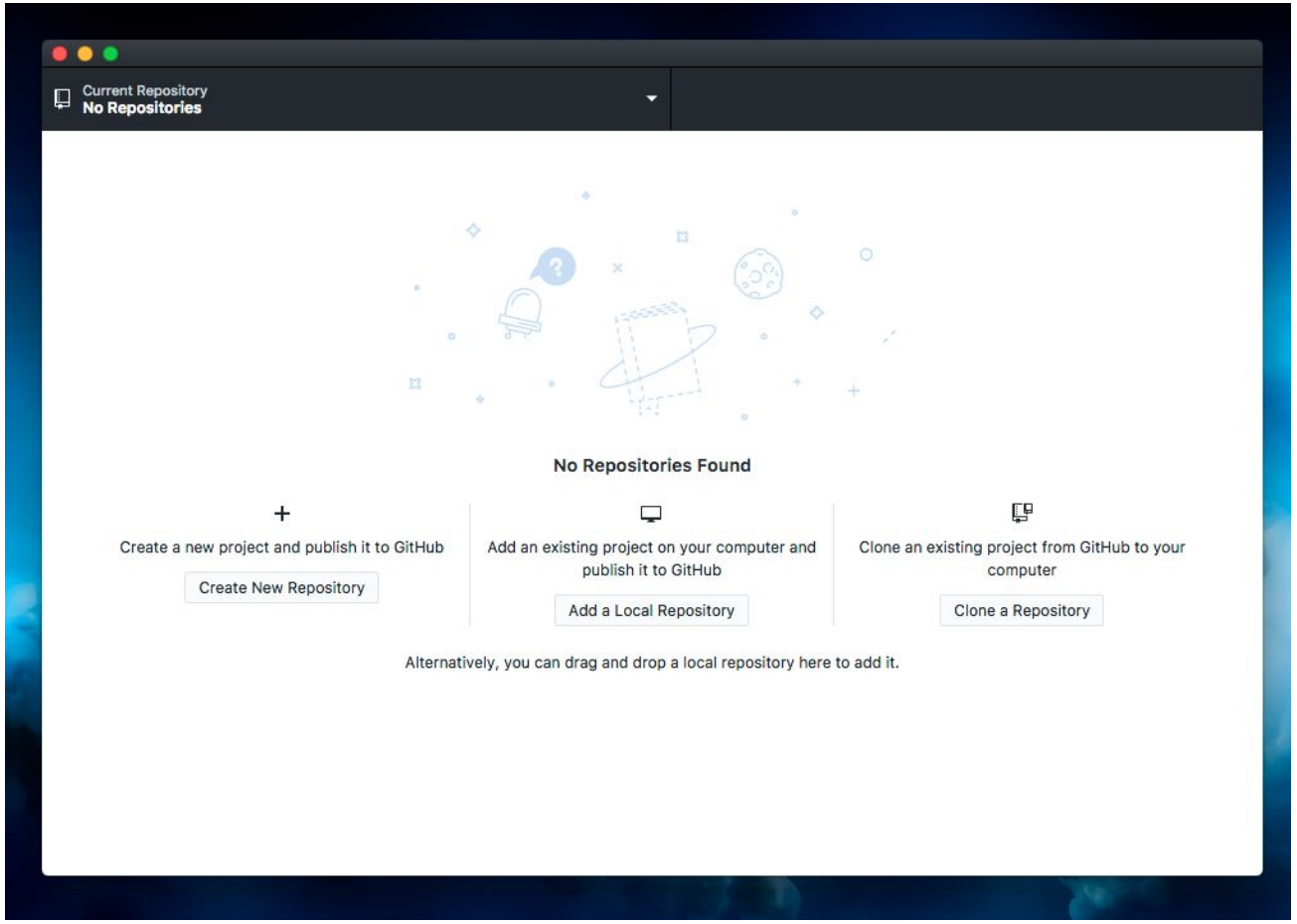
CodeLab II (CCO5000-20)

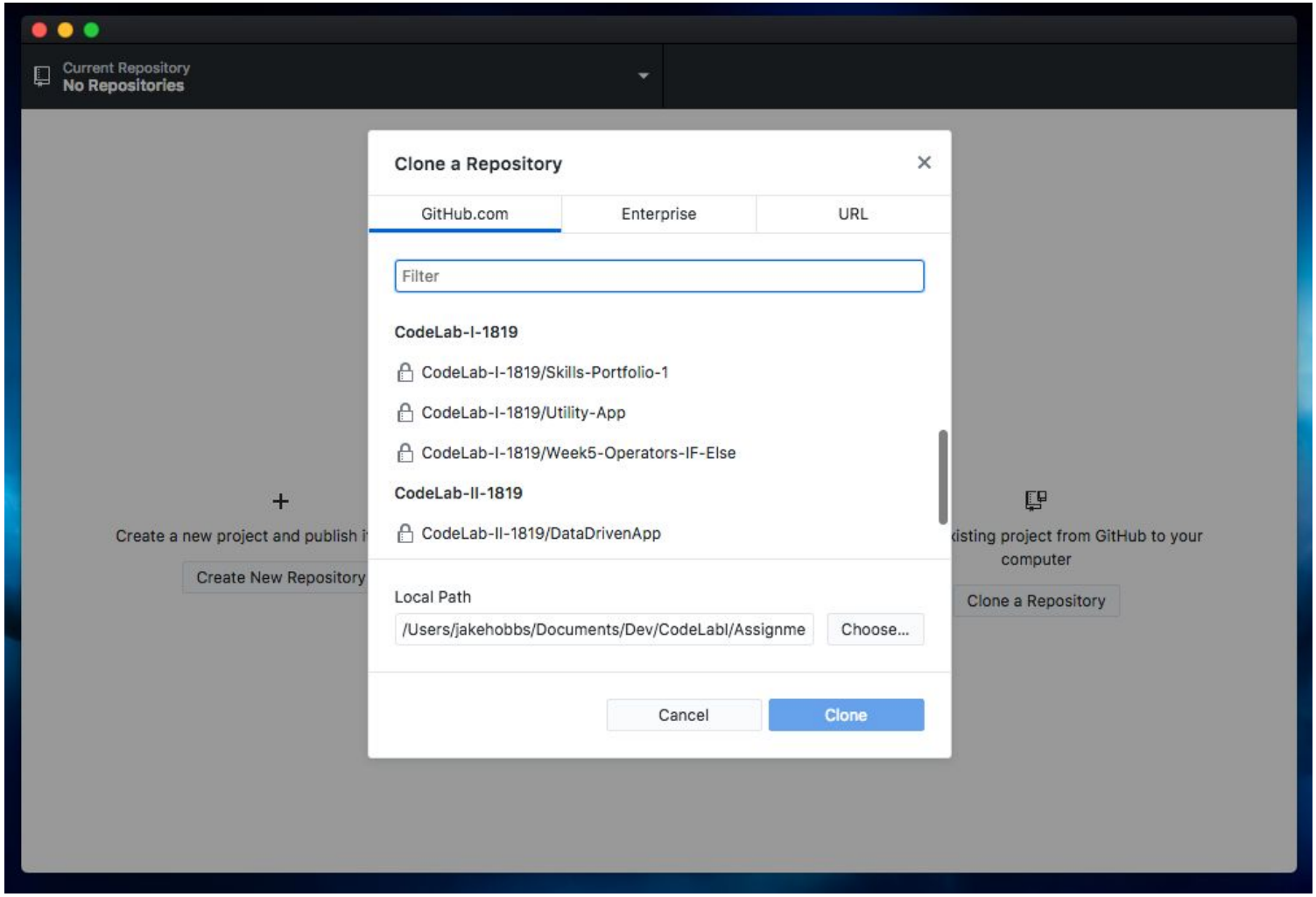
Cloning Option 1 Data Driven App Repository - Visual Studio

Creative Computing
The School of Creative Industries
College of Liberal Arts, Bath Spa University

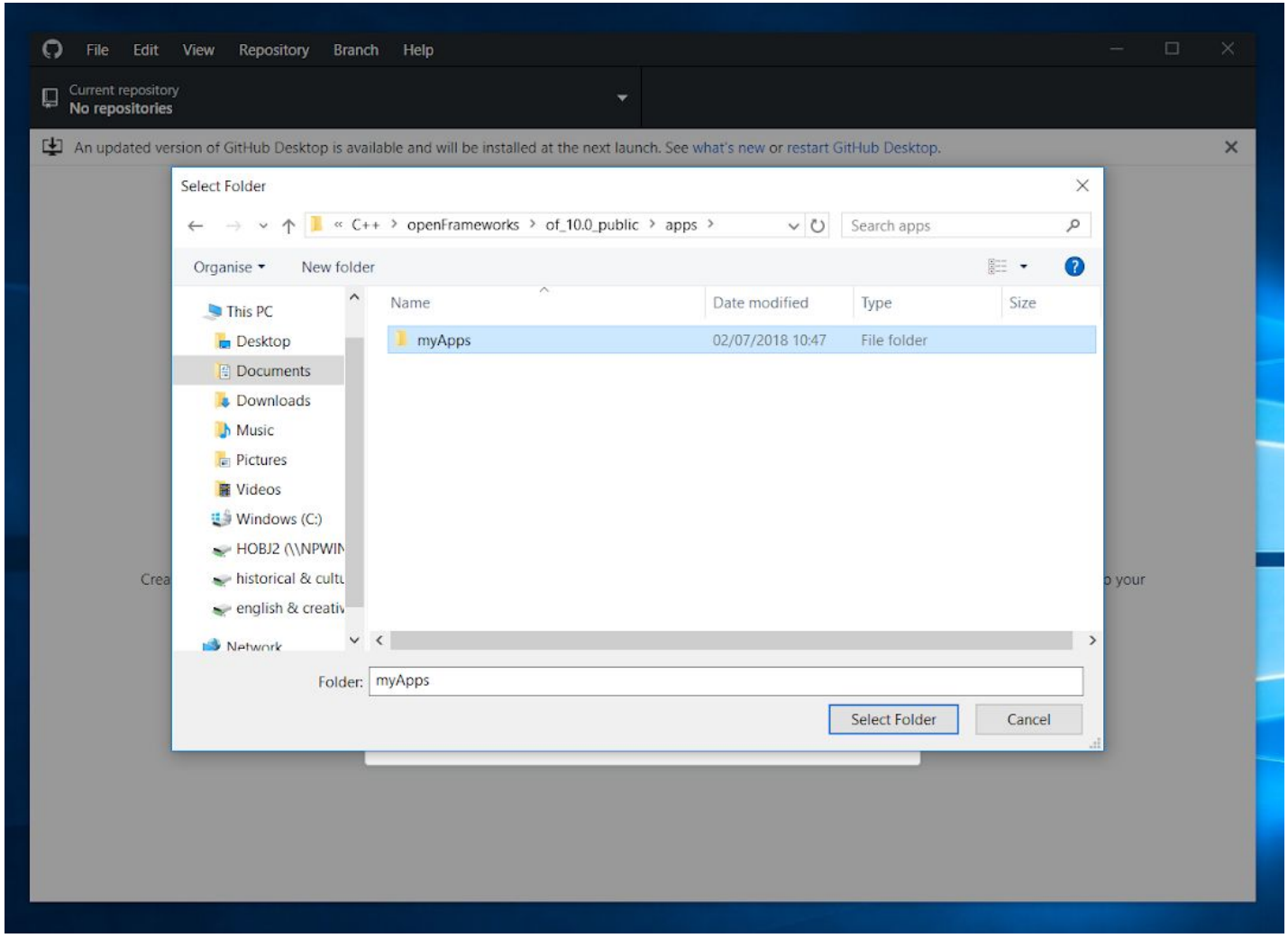
1. Use Github desktop clone your the repository to your computer. If you have no existing repositories you can do this by clicking the *“Clone a Repository”* button on the opening screen. Else simply go “*File → Clone Repository”*



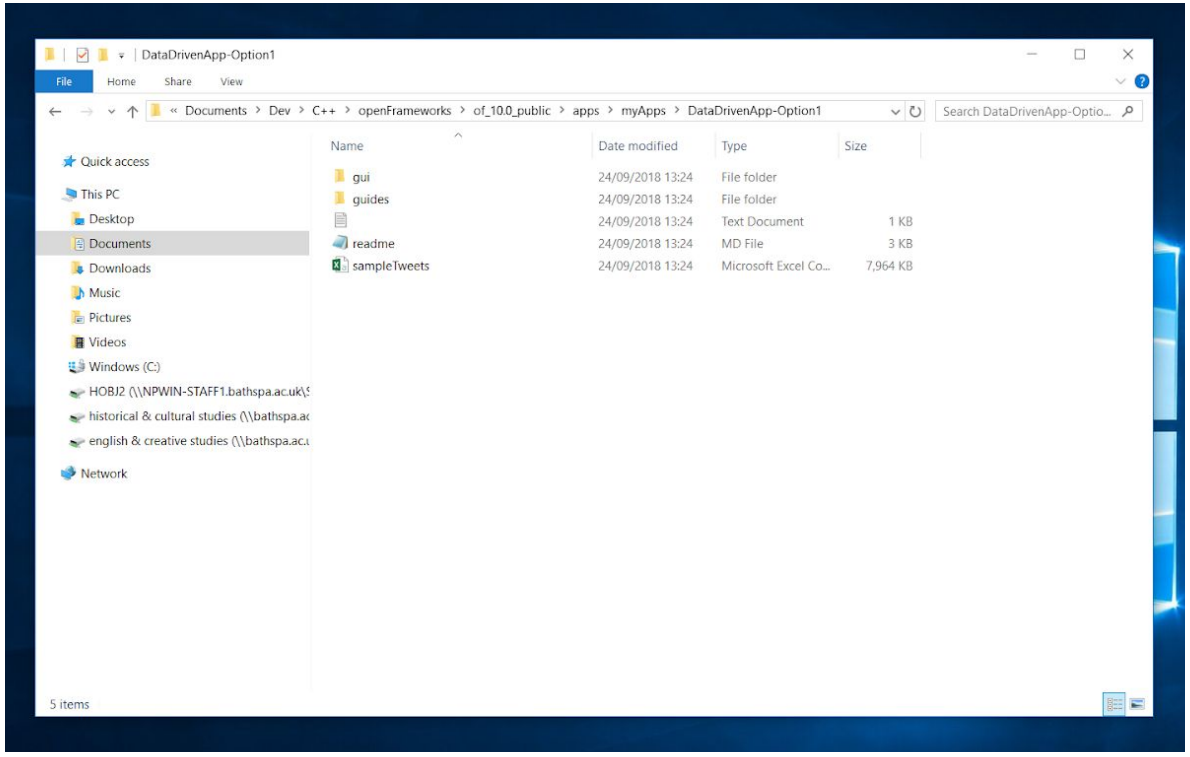
2. You will be presented with a screen similar to the one below. You should look for and select the assignment repository from those listed. This will be named something like: *CodeLab-II-1819/DataDrivenApp-Option1-yourgithubusername*



3. Click “*Choose...”* next to local path and browse to the “*myApps*” folder of your openFrameworks installation. If you do not have a *“myApps”* folder, create one. Click *“Select Folder”* and then “*Clone”* this will clone the repository.

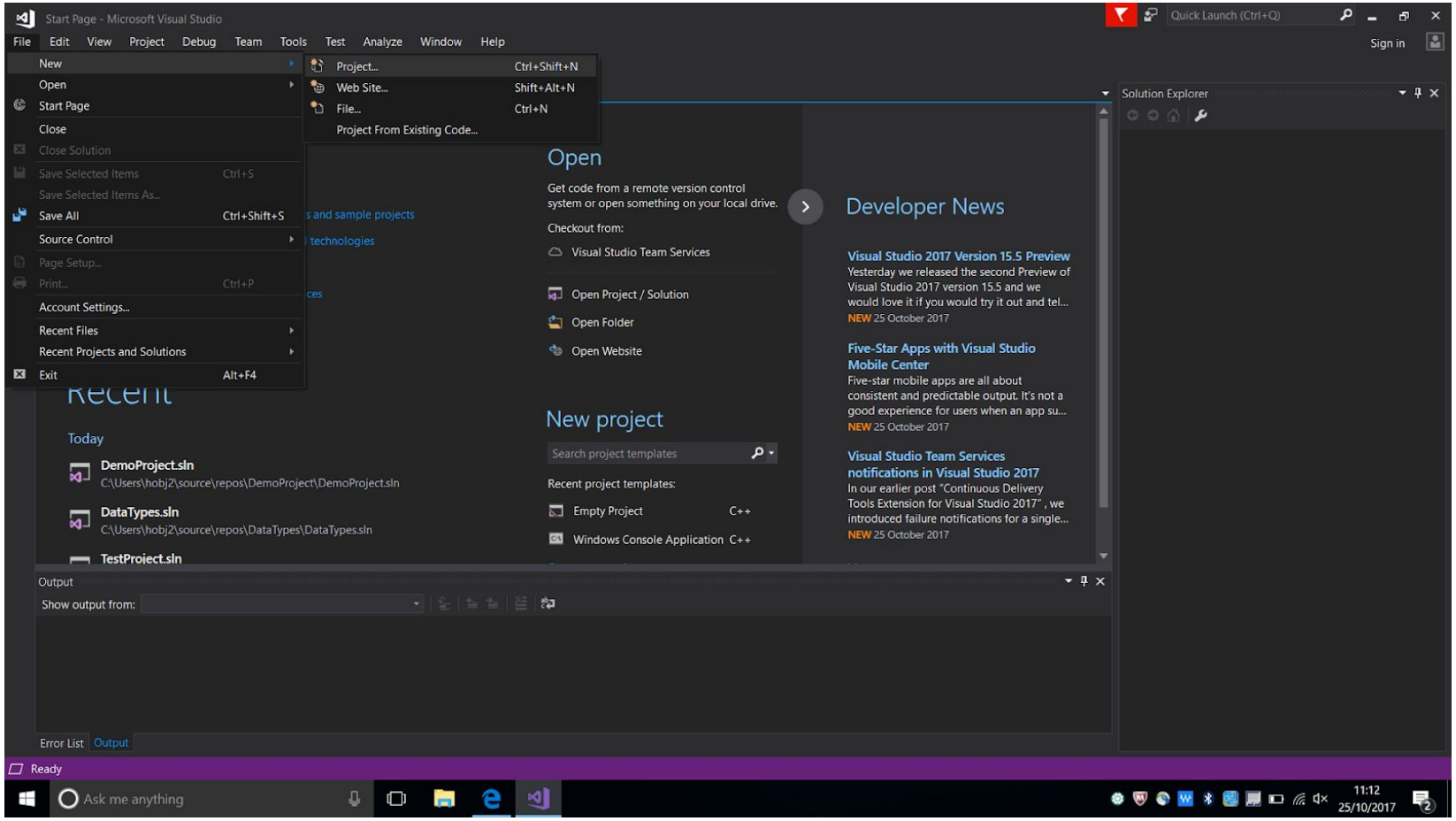


4. You should now have a copy of the repository from Github on your local machine. The image below a cloned repository after downloading to my machine. For the assignment you should create one project for your main console based application and a seperate project for your mock gui. The remaining steps guide you on how to create these projects.

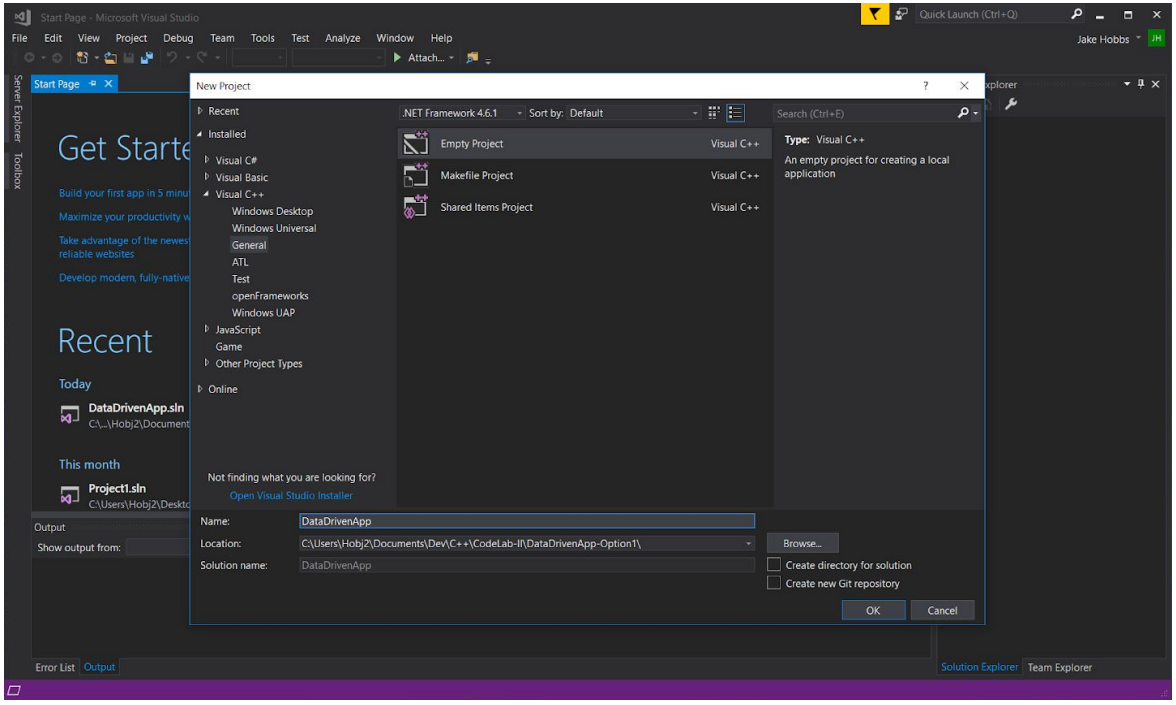


Setting up main console project

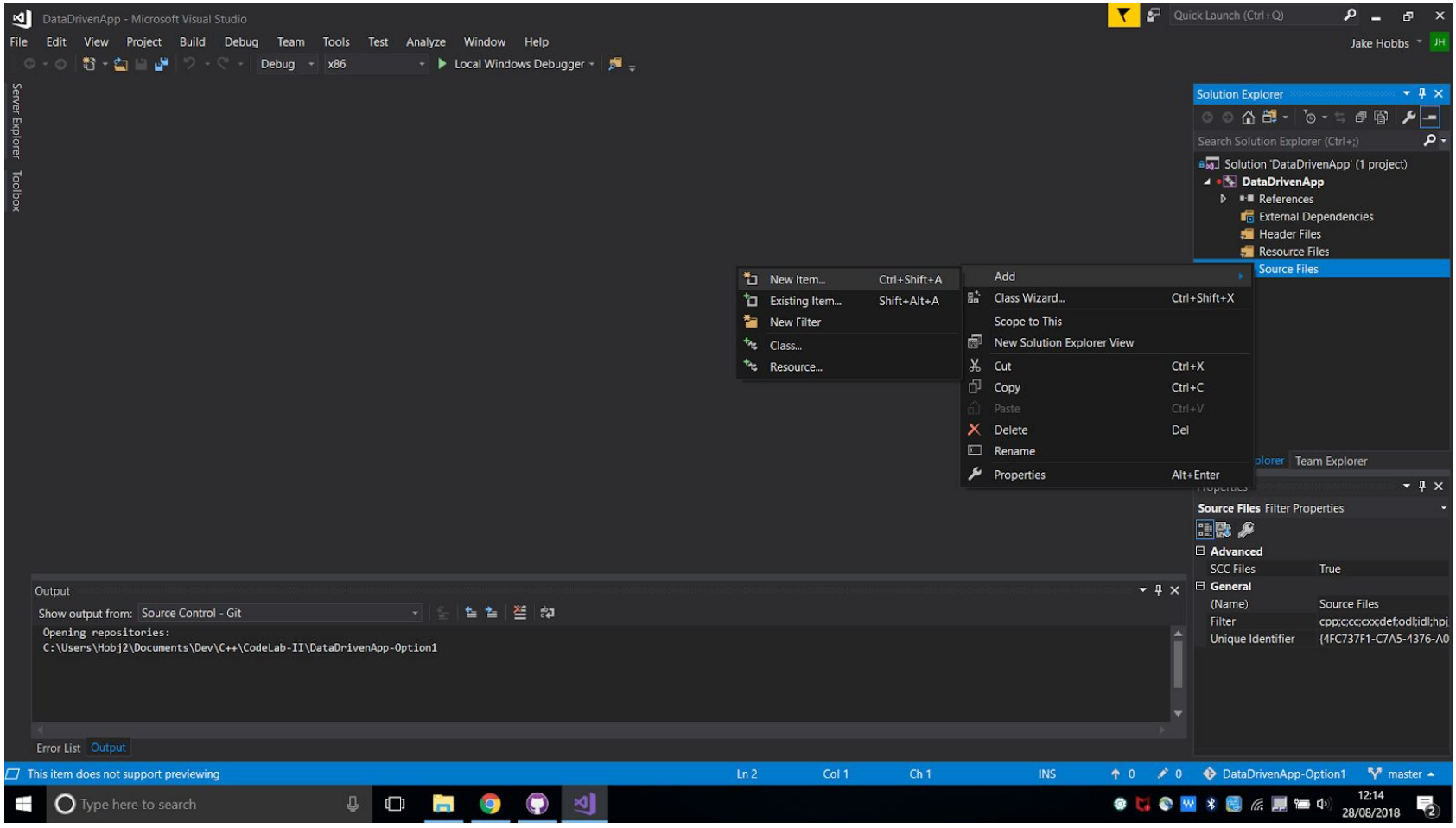
5. In Visual Studio go to File → New Project



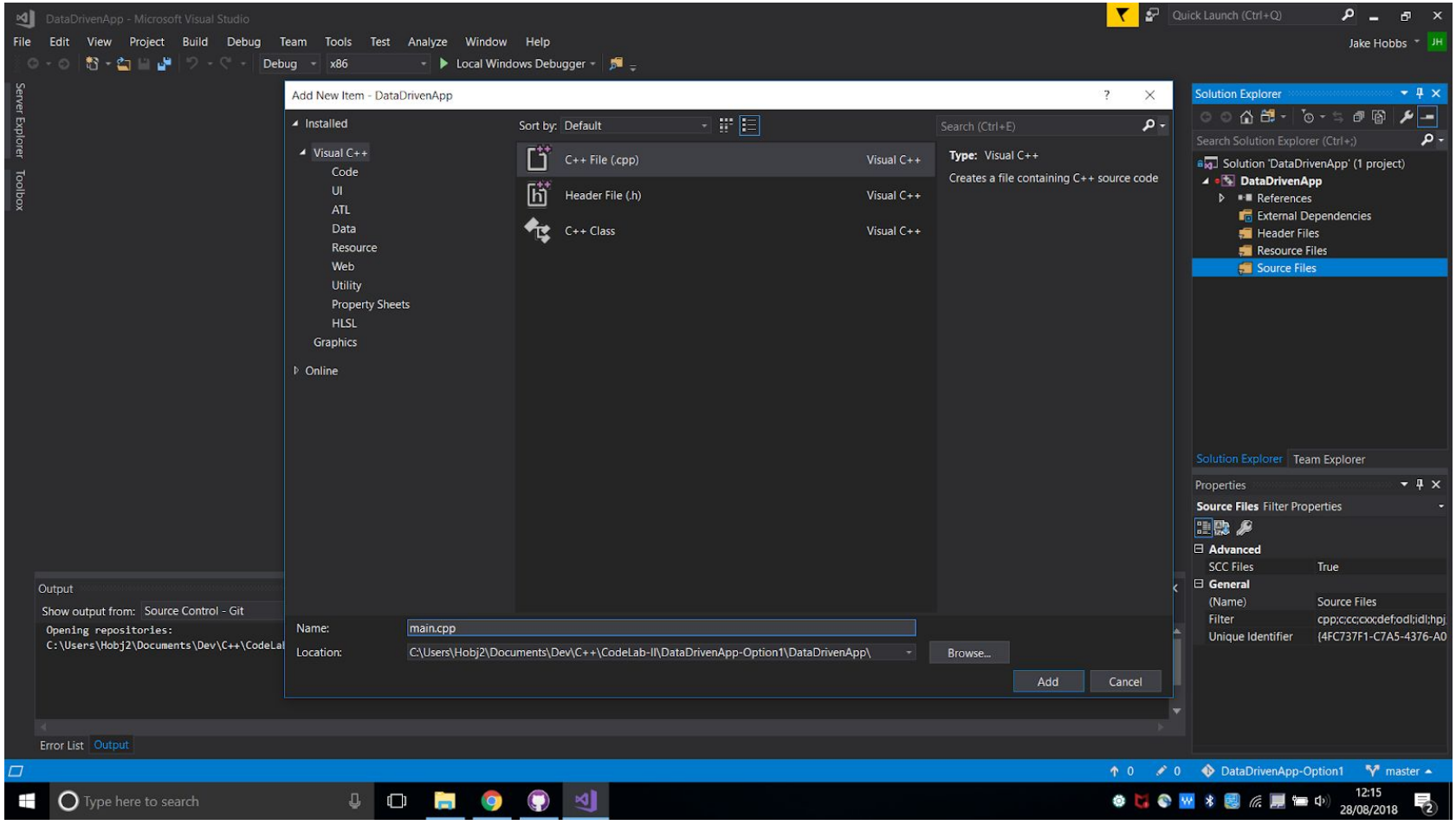
6. Create your project:
    a. Select *General* from under *Visual C++* on the left of the new project window
    b. Select *Empty Project* in the main window
    c. In the *"Name"* box provide a name for the project (e.g: *DataDrivenApp*)
    d. In the *"Location"* box click *"Browse..."* and select the location you cloned the repository to in step 3
    e. Make sure *"Create directory for Solution"* and *"Create new Git repository"* are **unchecked**.
    f. Click OK



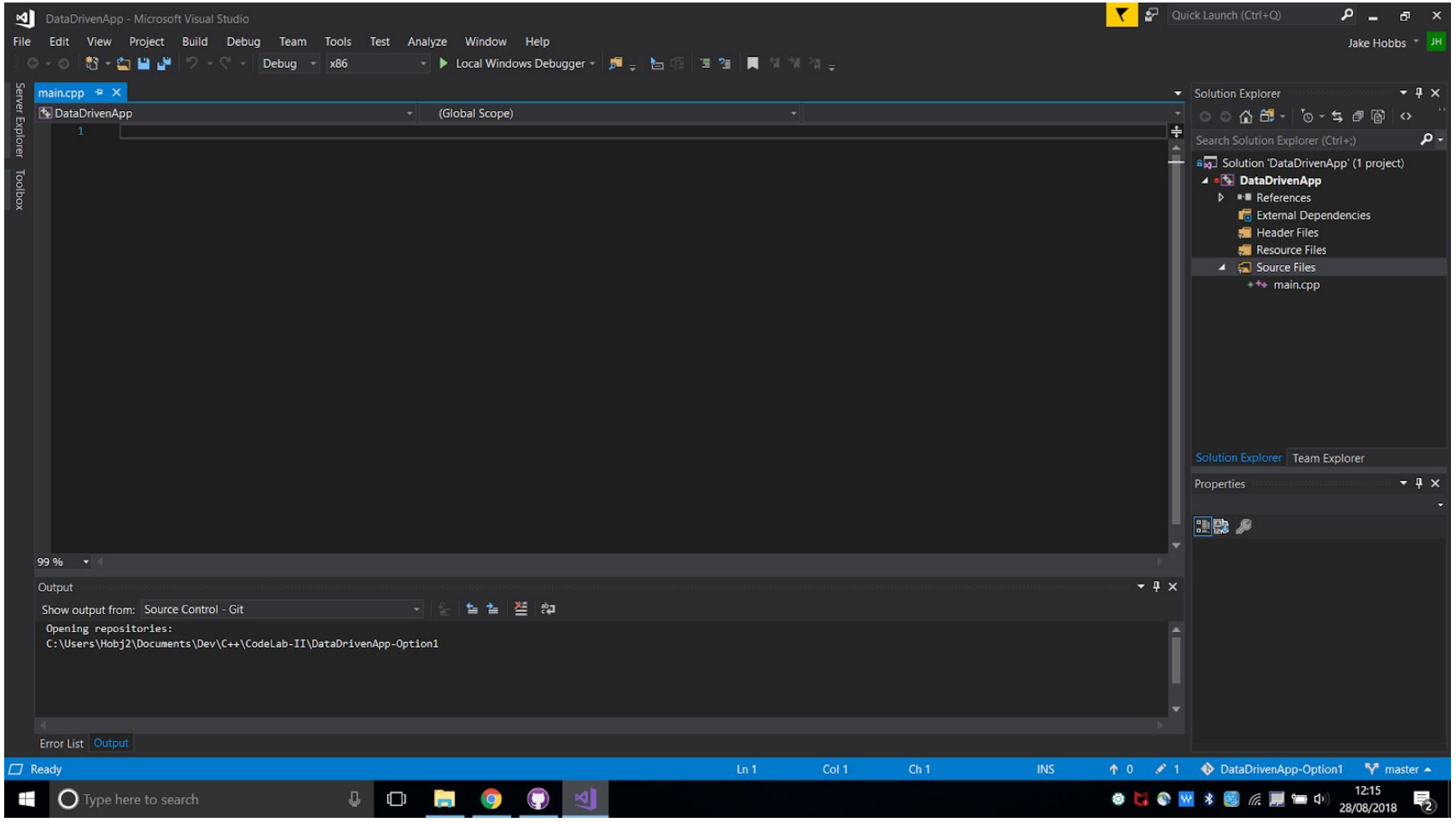
7. Your Visual Studio project is almost ready. You will be presented with the following screen. (Solution Explorer may appear on the left or the right).
    a. In the *Solution Explorer* window locate and right click the *Source Files* folder
    b. Click *Add → New Item* from the popup that appears



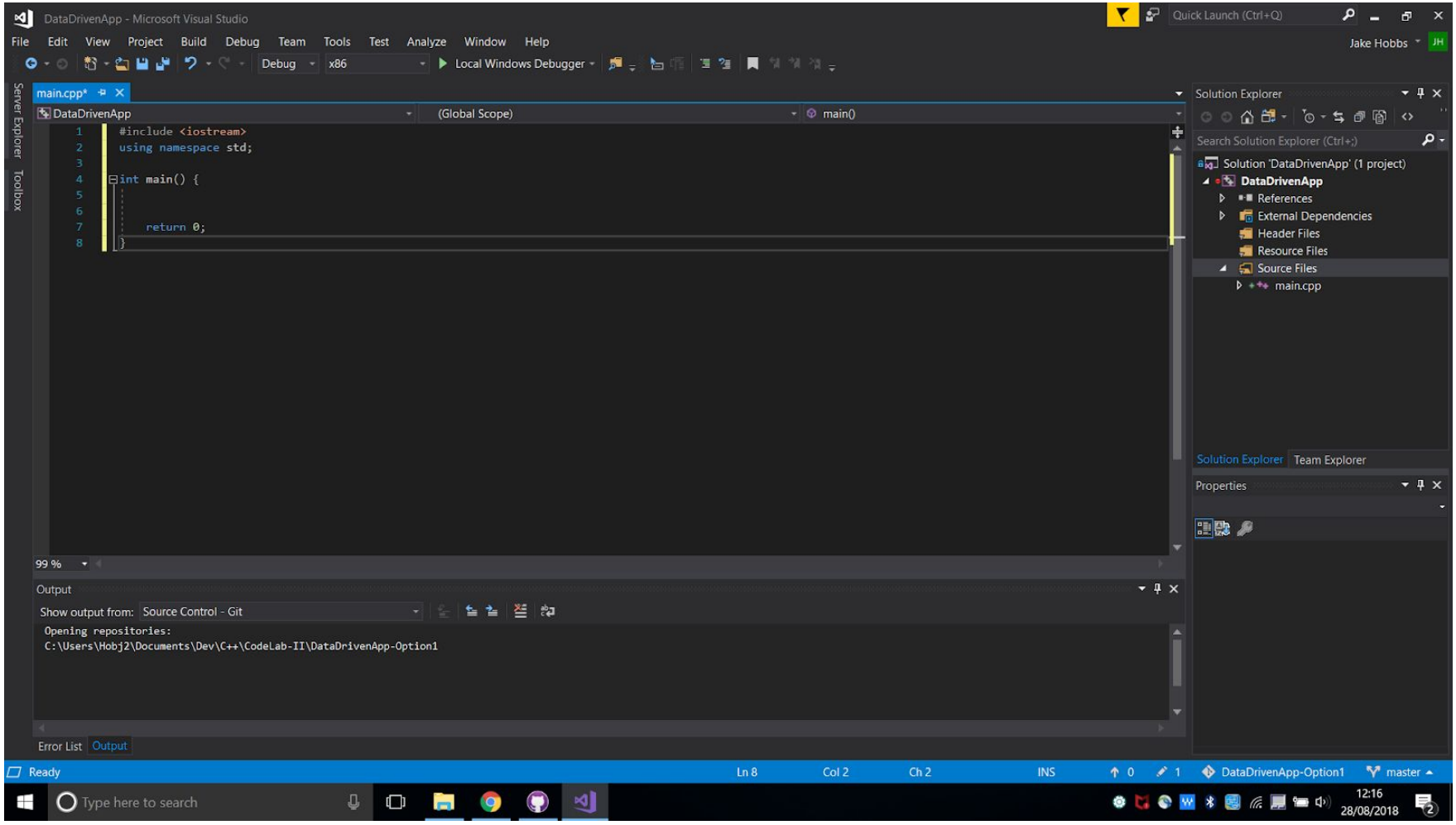
8. Add a new C++ (.cpp) file and name it main.cpp. Click add.



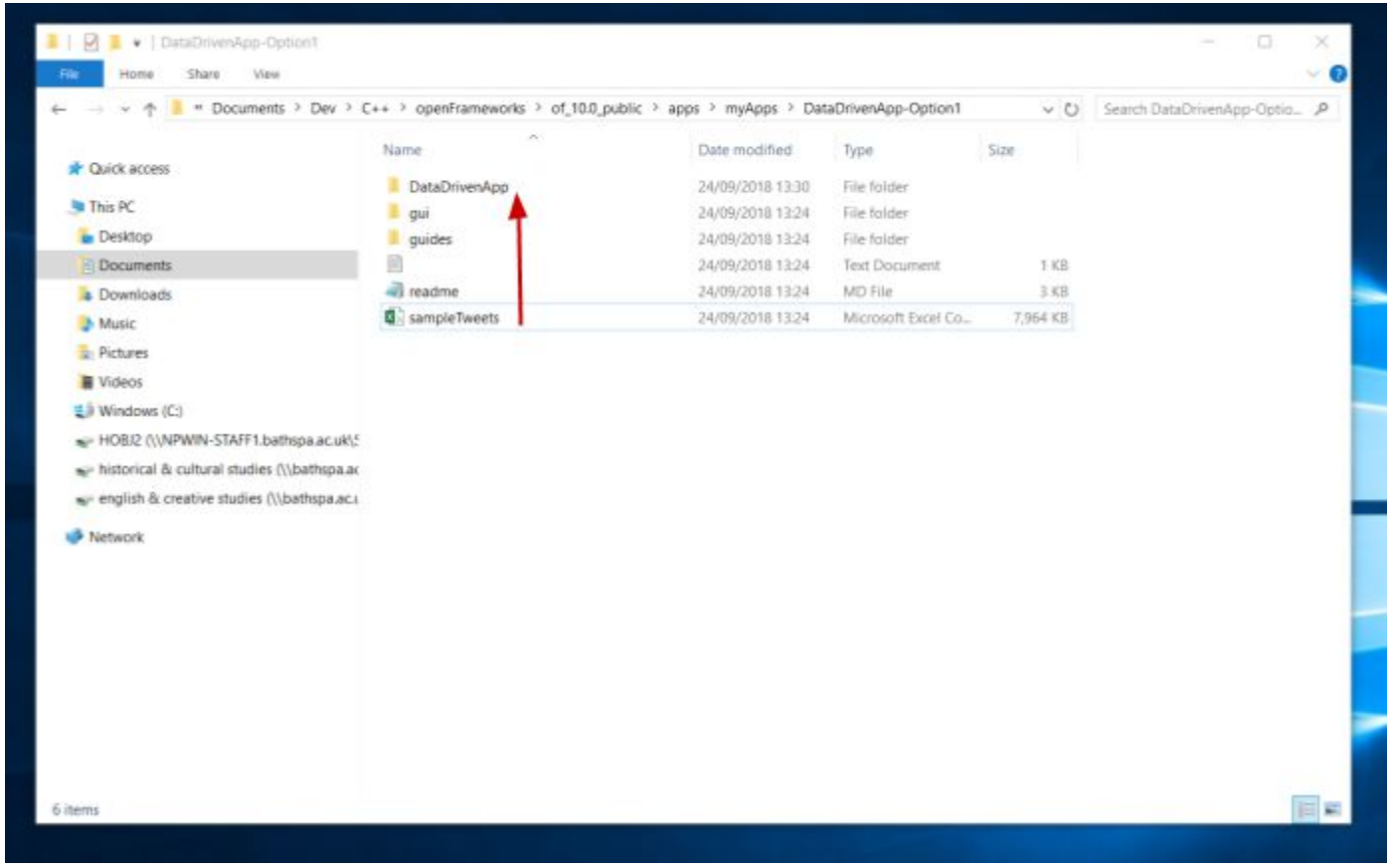
9. Your main.cpp file will initially be blank

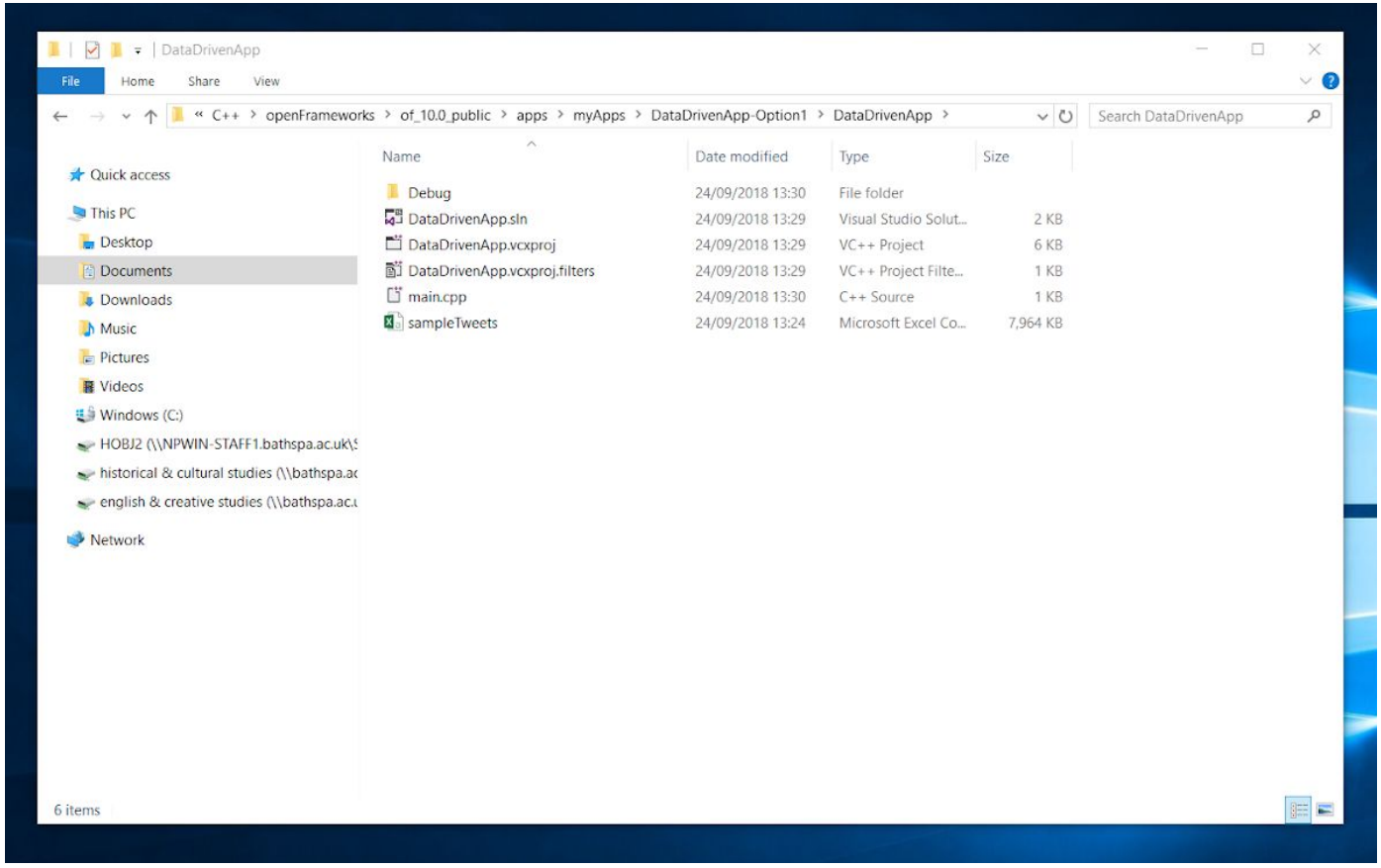


10. Add the following basic code to the file. You are now ready to code.



11. Go back to the folder on your computer where you cloned the repository to in step 3. Ensure the sampleTweets.csv file is in the same location as your main.cpp file.

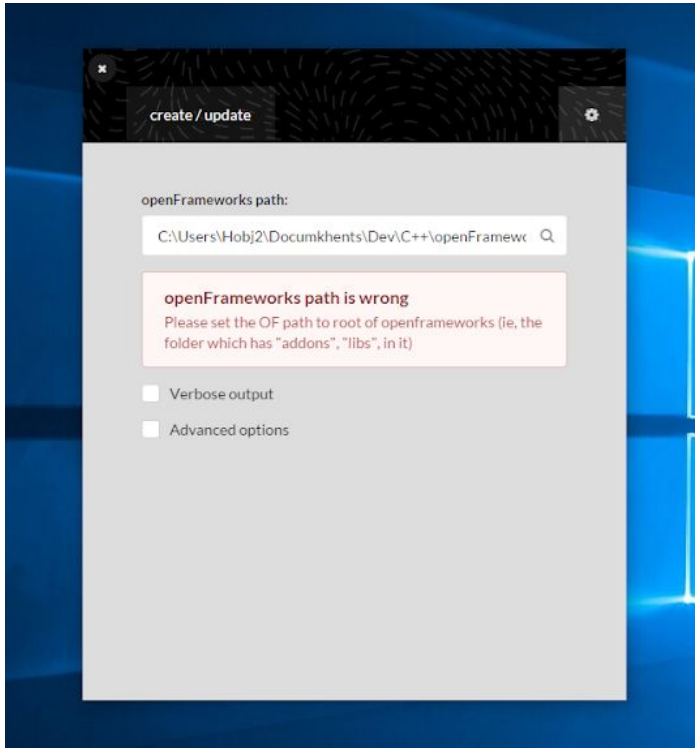


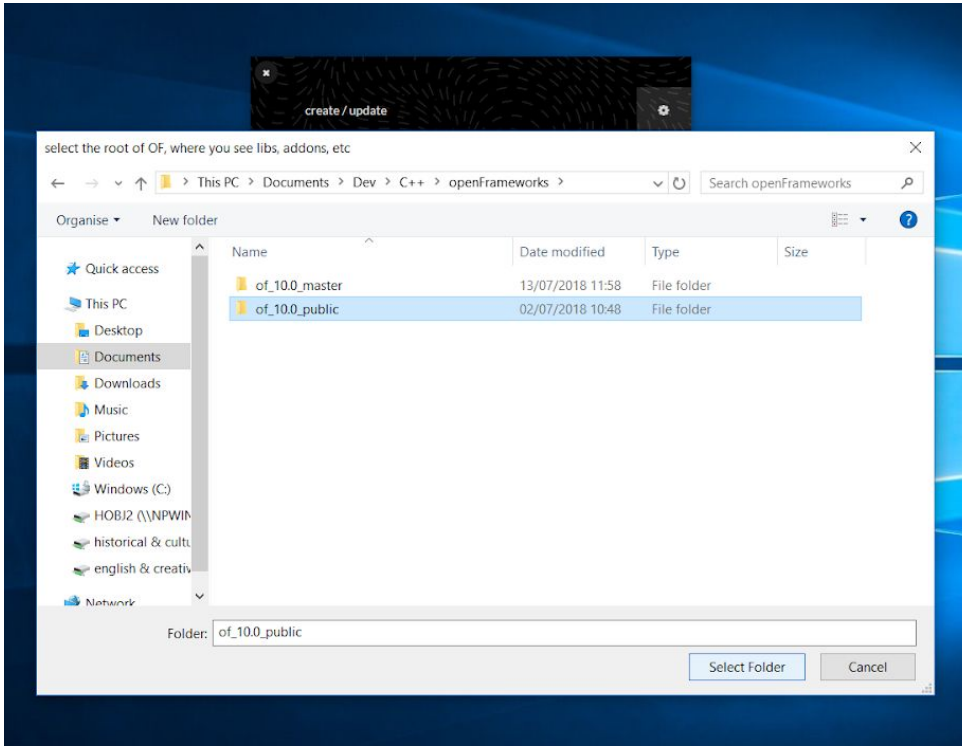

DataDrivenApp
File
Home
Share
View
« C++ › openFrameworks › of_10.0_public › apps › myApps › DataDrivenApp-Option1 › DataDrivenApp ›
Search DataDrivenApp
Quick access
This PC
Desktop
Documents
Downloads
Music
Pictures
Videos
Windows (C:)
HOBJ2 (\\NPWIN-STAFF1.bathspa.ac.uk\
historical & cultural studies (\\bathspa.ac
english & creative studies (\\bathspa.ac.
Network
Name
Date modified
Type
Size
Debug 24/09/2018 13:30 File folder
DataDrivenApp.sln 24/09/2018 13:29 Visual Studio Solut... 2 KB
DataDrivenApp.vcxproj 24/09/2018 13:29 VC++ Project 6 KB
DataDrivenApp.vcxproj.filters 24/09/2018 13:29 VC++ Project Filte... 1 KB
main.cpp 24/09/2018 13:30 C++ Source 1 KB
sampleTweets 24/09/2018 13:24 Microsoft Excel Co... 7,964 KB
6 items

Setting up mock gui project

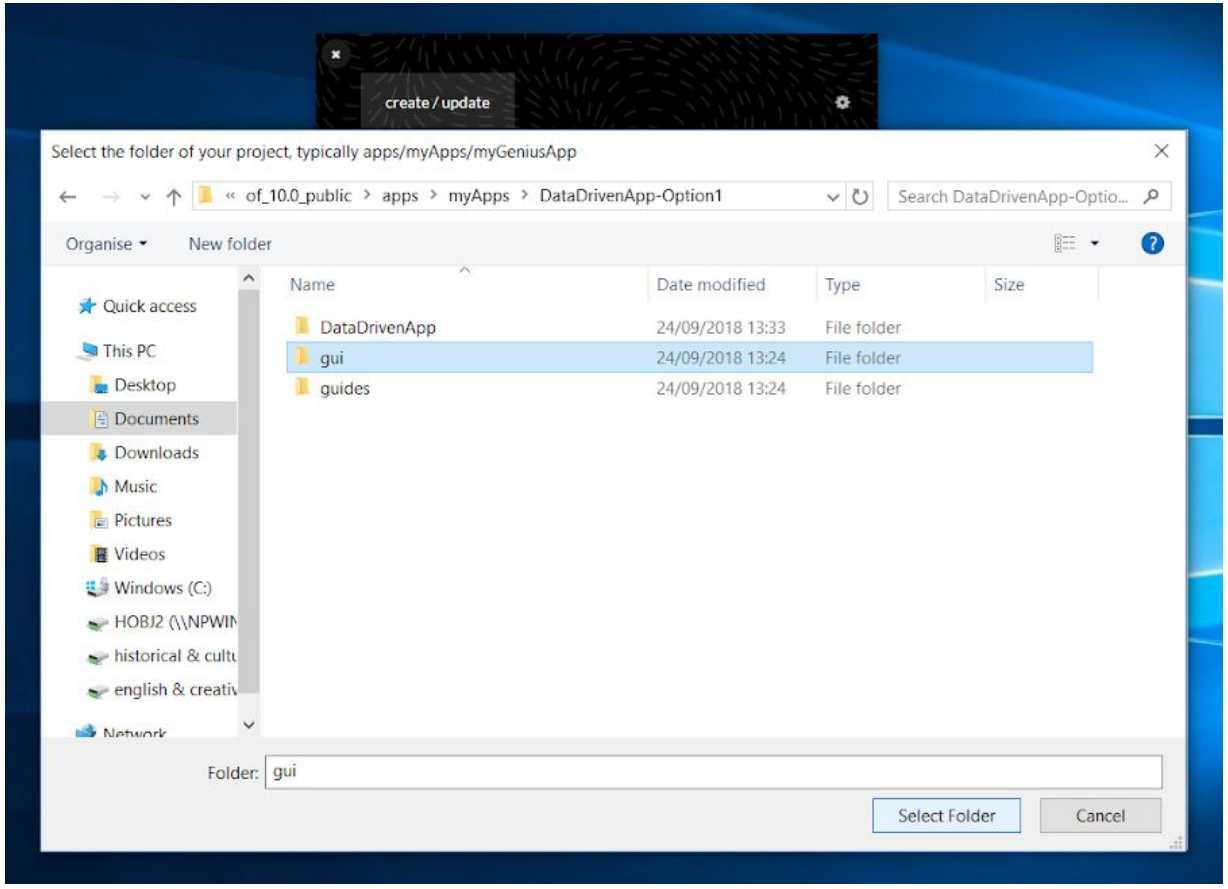
1. The openframeworks project can be created by importing the gui folder in the cloned repository via the project generator. This folder contains template code files for the standard main.cpp, ofApp.cpp and ofApp.h for a openFrameworks project. This can be done by following the remaining steps

2. If asked set the root path of your openFramworks installation in the project generator.

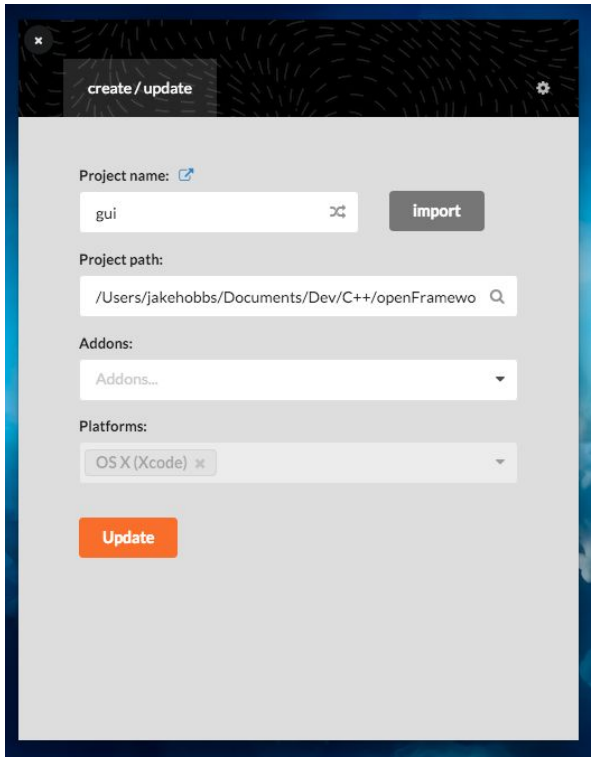




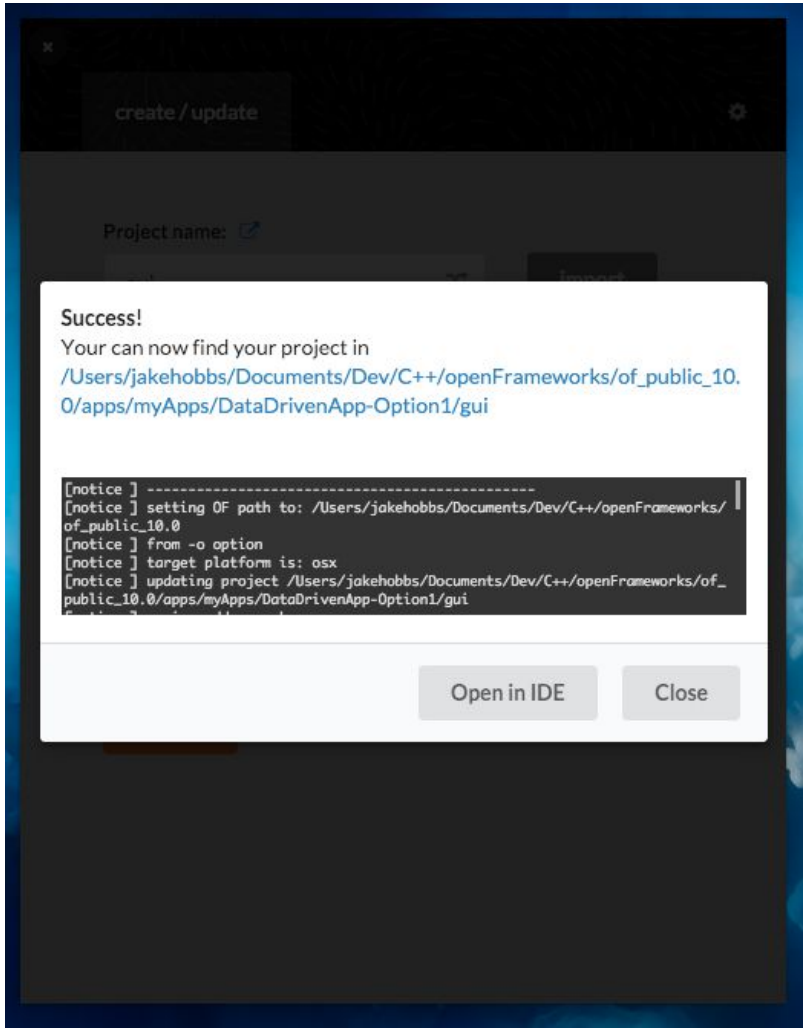
3. On the *“Create/Update”* tab select *“Import”* and locate the *“gui”* folder included in the repository. Select this folder and click “*Select Folder”.*



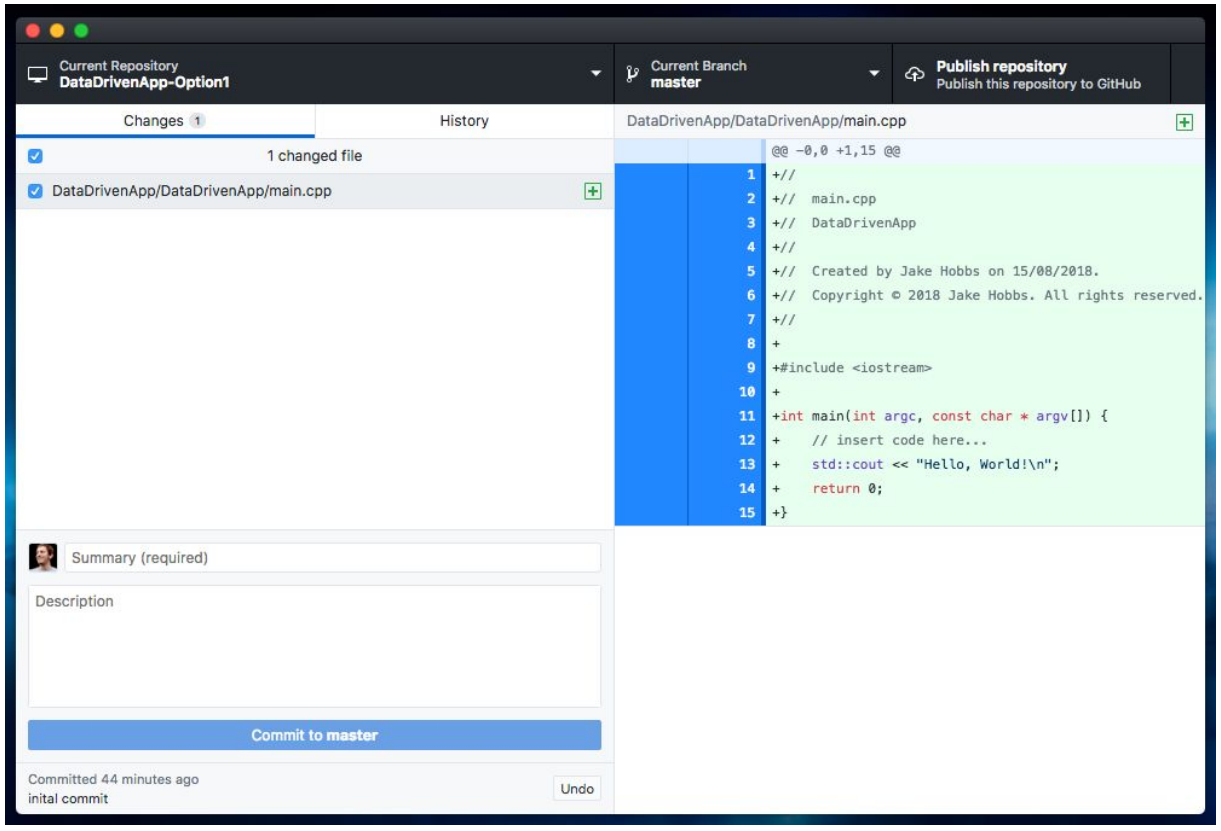
4. Click *“Update”*



5. Click *“Open in IDE”*



6. Your project is now ready to go. Github Desktop will keep track of your changes in your console project and the openFrameworks gui project. You should regularly make commits to ensure you can fall back to previous versions if things go wrong. You should also push your commits back up to your Github repository often, this will make sure you have a backup of your work.



7. When you are happy with your final solution make sure you make one final commit and push to your Github repository. Only code submitted to your assignment repository will be marked.